UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/625,812	07/26/2000	Timothy J. Van Hook	00100.00.0105	8263

29153 7590 08/11/2008

ADVANCED MICRO DEVICES, INC.
C/O VEDDER PRICE P.C.
222 N.LASALLE STREET
CHICAGO, IL 60601

EXAMINER	
HSU, JONI	

ART UNIT	PAPER NUMBER
2628	

MAIL DATE	DELIVERY MODE
08/11/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	09/625,812	VAN HOOK, TIMOTHY J.	
	Examiner	Art Unit	
	JONI HSU	2628	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 June 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-18,23,25-31 and 33* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *25-31* is/are allowed.

6)☒ Claim(s) *1-18,23 and 33* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

1. Applicant's arguments, see p. 9, filed June 11, 2008, with respect to the objections and 35 U.S.C. 112 rejections have been fully considered and are persuasive. The objections to Claims 15-18 and the 35 U.S.C. 112 rejections of Claim 17 have been withdrawn.

2. Applicant's arguments filed June 11, 2008, with respect to the 35 U.S.C. 103(a) rejections have been fully considered but they are not persuasive.

3. As per Claim 1, Applicant argues cited portion of column 8 in Joy (US006341347B1) refers to single processor and describes single processor does not switch threads to execute an otherwise unused or idle pipeline stage as alleged. The thread is switched, not the pipeline. The office action refers to unused or idle pipeline stage and confuses a pipeline stage with a plurality of pipelines on multiple processors. There is no mention of the number of programs that are interleaved to be greater than or equal to the depth of the pipeline as there is no discussion of the depth of the pipeline for a given shared pipeline in Joy. Applicant requests a showing as to the teaching of the actual depth of the pipeline versus the number of threads (p. 10).

In reply, the Examiner points out that Claim 1 recites "interleaving instructions from said plurality of programs and providing said instructions to said pipeline". Therefore, the instructions are switched. Claim 1 does not recite anything about pipelines being switched. Claim 1 also does not recite anything about a plurality of pipelines on multiple processors. Joy teaches switching threads or instructions (c. 7, ll. 41-44; c. 8, ll. 14-26; c. 10, ll. 14-37; c. 37, ll. 9-23), as recited in Claim 1. Joy teaches each pipeline stage selects one active thread (c. 7, ll. 41-44; c. 8, ll. 14-26; c. 10, ll. 14-37; c. 37, ll. 9-23), and so this means there are a same number of pipeline

stages as there are threads (programs). So, the execution pipeline has a depth less than or equal to the plurality of programs, as recited in Claim 1.

4. As per Claim 14, Applicant argues Krishna (US006161173A1) teaches that normal execution of an instruction from a single program is carried out with no no-op being inserted. Applicant claims a different operation (p. 11).

In reply, the Examiner points out that Joffe (US006330584B1) teaches interleaving instructions from N programs in processor pipeline (160, Fig. 1; c. 2, ll. 29-34; c. 3, ll. 40-42); and executing instructions such that a first instruction from one of the N programs is completed before beginning execution of a second instruction of the one of the N programs (c. 2, ll. 35-39). Krishna teaches scheduler allocates fixed latency, which is typically one clock cycle, between issuing instruction to pipeline and pipeline returning result (c. 4, ll. 1-4). For some instructions, pipeline has longer latency (c. 4, ll. 4-5). Since fixed latency is typically one clock cycle for one instruction, Krishna is considered to teach execution pipeline has average latency of one instruction per cycle. This would be obvious for reasons for Claim 1. Applicant's disclosure describes inserting no-ops into instruction stream or retarding launching of new programs until 1st program finishes (p. 17, ll. 8-11). So, when no no-op is inserted, that means 1st instruction is completed and execution of 2nd instruction can begin. Krishna teaches local scheduling circuitry stops main scheduler from issuing selected operation if latency of another operation would create conflict with main scheduler issuing selected operation (c, 2, ll. 56-60). So, it is ensured 1st instruction is completed before beginning execution of 2nd instruction. So, next instruction is not provided to the pipeline until a previous instruction has completed. Operation is executed if no no-op is inserted into pipeline (c. 5, ll. 36-38; c. 2, ll. 41-45). So, when no no-op is inserted into

pipeline, this ensures first instruction is completed before beginning execution of second instruction. So, when no-op is inserted, this means that operation is not ready to be executed. When associated operation which is to be executed is inserted, there is no no-op inserted, this means that associated operation is ready to be executed, meaning 1st instruction is completed and it is now okay to execute associated operation. So, no no-op is inserted for purpose of ensuring 1st instruction is completed before beginning execution of 2nd instruction. So combination of Joffe and Krishna does teach Claim 14.

5. As per Claim 33, Applicant argues that Joffe does not refer to whether a plurality of identified programs have been completed, but refers to the fact that a resource is not provided to a task until after every other task sharing the resource has finished accessing the resource. The task may still be uncompleted but the resource is allocated (p. 12).

In reply, Examiner points out Joffe teaches if task attempts to access unavailable resource, task is suspended. When resource becomes available, suspended task is resumed, and instruction accessing resource is re-executed. Task does not get access to same resource until after every other task sharing resource has finished accessing resource (c. 2, ll. 29-39). If Wait signal is asserted, instruction execution is not completed and PC register is frozen, but task remains active, and instruction will be executed again starting next clock cycle. If Suspend/Wait signals are deasserted, PC register is changed to point to next instruction (c. 10, ll. 19-32). So, Joffe teaches checking to see if Suspend/Wait signals are desasserted, which indicates instruction execution is completed (c. 10, ll. 19-32) and resource has become available (c. 2, ll. 32-34), and resource becomes available after every other task sharing resource has finished accessing resource (c. 2, ll. 29-39). So, Joffe teaches checking to see if all of the programs are completed.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

8. Claims 1-7, 9-11, and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Joy (US006341347B1) in view of Krishna (US006161173A).

9. As per Claim 1, Joy teaches programmable processor for executing plurality of programs (threads), each of plurality of programs has plurality of instructions (c. 8, ll. 33-39). Multiple-thread execution pipeline includes pipeline stages. Each pipeline stage includes flip-flops, and each flip-flop is coupled to select-bus lines selecting active thread from among plurality of execution threads. This allows each pipeline stage to immediately switch from first thread to second thread when first thread stalls, so second thread is executing on otherwise unused or idle pipeline stage. This allows for execution of more threads without increasing number of pipeline stages (c. 7, ll. 41-44; c. 8, ll. 14-26; c. 10, ll. 14-37; c. 37, ll. 9-23). Since each pipeline stage selects one active thread, this means there are a same number of pipeline stages as there are

threads (programs). So, programmable processor has execution pipeline having depth less than or equal to plurality of programs. Since each pipeline stage switches from executing instructions from first thread to executing instructions from second thread when first thread stalls, and later resuming execution of instructions of postponed stalling first thread (c. 3, ll. 14-25; c. 7, ll. 41-45; c. 8, ll. 27-39), this means there is interleaver for interleaving instructions from plurality of programs and providing instructions to pipeline for execution such that number of plurality of programs that are interleaved is greater than or equal to depth of pipeline.

However, Joy does not explicitly teach execution pipeline has average pipeline latency of one instruction per cycle. However, Krishna teaches scheduler allocates fixed latency, which is typically one clock cycle, between issuing instruction to execution pipeline and execution pipeline returning result (c. 4, ll. 1-4). For some instructions, execution pipeline has longer latency (c. 4, ll. 4-5). Since fixed latency is typically one clock cycle for one instruction, Krishna is considered to teach execution pipeline has average latency of one instruction per cycle.

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify device of Joy so pipeline has average latency of one instruction per cycle as suggested by Krishna because it results in a more streamlined pipeline operation and simplified design (Krishna, c. 2, ll. 60-67). Even though Joy does not teach pipeline (Joy, c. 37, ll. 9) has average latency of one instruction per cycle, Krishna teaches it is typical for instructions in pipeline to have average latency of 1 instruction per cycle. So, it would be obvious that pipeline of Joy can be used to execute instructions that have average latency of 1 instruction per cycle.

10. As per Claim 2, Joy teaches that the pipeline has a datapath with a depth equal to the number of programs (c. 10, ll. 14-37; c. 37, ll. 9-23).

11. As per Claim 3, Joy does not teach next instruction from one program is not provided to pipeline until previous instruction of one of the programs has completed. But, Krishna teaches local scheduling circuitry stops main scheduler from issuing selected operation if latency of another operation would create conflict with main scheduler issuing selected operation (c. 2, ll. 56-60). So, next instruction is not provided to pipeline until previous instruction has completed.

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify device of Joy so next instruction from one of plurality of programs is not provided to pipeline until previous instruction of one of plurality of programs has completed because Krishna suggests sometimes latency of another instruction would create conflict with main scheduler issuing selected instruction, so in order to avoid this conflict, next instruction is not provided to pipeline until previous instruction has completed (c. 2, ll. 56-60).

12. As per Claim 4, Joy teaches each program of the plurality of programs is independent of the other of the plurality of programs (c. 3, ll. 14-25).

13. As per Claim 5, Joy teaches interleaving instructions (c. 3, ll. 14-25; c. 7, ll. 41-45; c. 8, ll. 27-39), and so instructions are executed out of order.

However, Joy does not teach output buffer for storing out of order data output. However, Krishna teaches execution engine (140, Fig. 1) has an out-of-order architecture (c. 5, ll. 11-12), and scheduler (150) receives results from execution units (170, 175, 180) and stores results (c. 5, ll. 28-35). So, Krishna inherently teaches output buffer for storing out of order data output.

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify device of Joy to include output buffer for storing out of order data output

because Krishna suggests since instructions are executed out of order (c. 5, ll. 11-12), output buffer is needed to store out of order data output so data can put in correct order (c. 5, ll. 28-35).

14. As per Claim 6, Joy discloses one or more of a register copy, program counter, and program counter stack provided for each of the plurality of programs (c. 6, ll. 34-36).

15. As per Claim 7, Joy teaches one of control/computing resources, instructions, instruction memory, data paths, data memory, caches are shared by plurality of programs (c. 8, ll. 59-66).

16. As per Claim 9, Joy teaches instructions for loading data from memory (c. 8, ll. 59-67).

17. As per Claim 10, Joy teaches instructions for storing data in memory (c. 8, ll. 59-67).

18. As per Claim 11, Joy discloses that the data memory comprises a cache (c. 8, ll. 59-67).

19. As per Claim 23, Joy teaches programmable processor for executing plurality of programs, programmable processor having execution pipeline having depth less than or equal to plurality of programs wherein each of plurality of programs has plurality of instructions; and interleaver for interleaving instructions from plurality of programs and providing instructions to pipeline for execution, as discussed in the rejection for Claim 1.

However, Joy does not teach execution pipeline has average pipeline latency of one instruction per cycle; and next instruction from one of plurality of programs is not provided to pipeline until a previous instruction of one of plurality of programs has completed and wherein no no-op is inserted into pipeline for purpose of ensuring next instruction is not provided to pipeline until previous instruction has completed. But, Krishna teaches scheduler allocates fixed latency, which is typically one clock cycle, between issuing instruction to pipeline and pipeline returning result (c. 4, ll. 1-4). For some instructions, pipeline has longer latency (c. 4, ll. 4-5). Since fixed latency is typically one clock cycle for one instruction, Krishna is considered to teach

execution pipeline has average latency of one instruction per cycle. This would be obvious for reasons for Claim 1. Applicant's disclosure describes inserting no-ops into instruction stream or retarding launching of new programs until 1st program finishes (p. 17, ll. 8-11). So, when no no-op is inserted, that means 1st instruction is completed and execution of 2nd instruction can begin. Krishna teaches local scheduling circuitry stops main scheduler from issuing selected operation if latency of another operation would create conflict with main scheduler issuing selected operation (c, 2, ll. 56-60). So, it is ensured 1st instruction is completed before beginning execution of 2nd instruction. So, next instruction is not provided to the pipeline until a previous instruction has completed. This would be obvious for reasons for Claim 3. Operation is executed if no no-op is inserted into pipeline (c. 5, ll. 36-38; c. 2, ll. 41-45). So, when no no-op is inserted into pipeline, this ensures first instruction is completed before beginning execution of second instruction. So, when no-op is inserted, this means that operation is not ready to be executed. When associated operation which is to be executed is inserted, there is no no-op inserted, this means that associated operation is ready to be executed, meaning 1st instruction is completed and it is now okay to execute associated operation. So, no no-op is inserted for purpose of ensuring 1st instruction is completed before beginning execution of 2nd instruction.

It would be obvious to one of ordinary skill in the art at the time of invention by applicant to modify Joy to include checking no no-op is inserted into pipeline for ensuring next instruction is not provided to pipeline until previous instruction has completed because Krishna suggests no-op is needed for indicating previous instruction has not yet completed (c. 5, ll. 26-38).

20. Claim 8 is rejected under 35 U.S.C. 103(a) as being unpatentable over Joy (US006341347B1) and Krishna (US006161173A) in view of Nguyen (US005961628A).

Joy and Krishna are relied upon for the teachings for Claim 1. Joy and Krishna implicitly teach SIMD execution of vector instructions without addressing vector lengths.

But, Joy-Krishna do not teach executing SIMD vector instructions of length N and executing in parallel instructions having SIMD lengths that sum up to N. However, Nguyen teaches processor executes SIMD vector instructions of vector length N and executes in parallel plurality of instructions having SIMD vector lengths that sum up to N (c. 1, ll. 11-24, 53-60).

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify Joy-Krishna to include executing in parallel instructions having SIMD vector lengths that sum up to N because Nguyen teaches fast speed for repetitive tasks (c. 1, ll. 10-25).

21. Claims 12 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Joy (US006341347B1) and Krishna (US006161173A) in view of Narayanaswami (US005973705A).

Joy and Krishna are relied upon for teachings as discussed above relative to Claim 9.

However, Joy-Krishna do not teach address space of data memory has frame buffer unit and texture memory unit. But, Narayanaswami teaches SIMD graphics processing system having frame buffer unit (frame buffer 110f, Fig. 2A) while implicitly suggesting texture memory unit.

It would have been obvious to one of ordinary skill in the art at the time of invention by applicant to modify Joy and Krishna so address space of data memory has frame buffer unit and texture memory unit because Narayanaswami teaches it reduces processing time (c. 2, ll. 20-22).

22. Claims 14-17 and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Joffe (US006330584B1) in view of Krishna (US006161173A)

23. As per Claim 14, Joffe teaches executing instructions from plurality of programs comprising identifying N programs of plurality of programs wherein each of the plurality of

programs has a plurality of instructions (c. 2, ll. 11-14, 66-67; c. 1, ll. 62-c. 2, ll. 7); interleaving instructions from N programs in processor pipeline (160, Fig. 1; c. 2, ll. 29-34; c. 3, ll. 40-42); and executing instructions such that a first instruction from one of the N programs is completed before beginning execution of a second instruction of the one of the N programs (c. 2, ll. 35-39)

But, Joffe doesn't teach pipeline has average latency of 1 instruction per cycle, checking no no-op is inserted into pipeline for purpose of ensuring 1st instruction is completed before beginning execution of 2nd instruction. But, Krishna teaches this, as discussed for Claim 23.

24. As per Claim 15, Joffe teaches assigning program counter to each program (c. 2, ll. 8-13).

25. As per Claim 16, Joffe teaches assigning register to each of N programs (c. 2, ll. 8-13).

26. As per Claim 17, Joffe teaches processor pipeline has depth of N (c. 1, ll. 62-65).

27. As per Claim 33, Joffe teaches executing instructions from plurality of programs (c. 2, ll. 66-67), assigning 1st output register slot to first of plurality of programs wherein each of the plurality of programs has plurality of instructions (c. 1, ll. 62-c. 2, ll. 11). If wait signal is asserted, instruction execution is not completed, so instruction will be executed again until wait signals are deasserted, then next instruction can be executed (c. 10, ll. 20-24, 31-32), and process repeats until all instructions have been executed. Joffe teaches if task attempts to access unavailable resource, task is suspended. When resource becomes available, suspended task is resumed, and instruction accessing resource is re-executed. Task does not get access to same resource until after every other task sharing resource has finished accessing resource (c. 2, ll. 29-39). If Wait signal is asserted, instruction execution is not completed and PC register is frozen, but task remains active, and instruction will be executed again starting next clock cycle. If Suspend/Wait signals are deasserted, the PC register is changed to point to next instruction (c.

10, ll. 19-32). So, Joffe teaches checking to see if Suspend/Wait signals are desasserted, which indicates instruction execution is completed (c. 10, ll. 19-32) and resource has become available (c. 2, ll. 32-34), and resource becomes available after every other task sharing resource has finished accessing resource (c. 2, ll. 29-39). So, Joffe teaches checking to see if all of the programs are completed. So, Joffe teaches executing instructions of first program until program is completed; loading output of first program into its reserved space when first program is completed (c. 9, ll. 26-41); checking to see if all of plurality of programs are completed (c. 2, ll. 35-39). Wait signal is asserted if register is not available, and wait signal is deasserted if register is available for new instruction (c. 10, ll. 20-24, 31-32). Each task (program) has separate register and separate flags (c. 2, ll. 11-13). So, 2nd output register slot is assigned to second program. If task attempts to access unavailable resource, task is suspended. When resource becomes available, suspended task is resumed, and instruction accessing resource is executed (c. 2, ll. 29-34). So, Joffe teaches checking to see if 2nd register slot is available to assign to 2nd program from plurality of programs when 1st program is completed; checking to see if one or more instructions are available when at least one of the programs is not completed (c. 2, ll. 35-39).

However, Joffe does not teach placing no-op when no more instructions are available. However, Krishna teaches information in each entry describes either no-op or associated operation which is to be executed (c. 5, ll. 36-38). So, when there is no-op, that means that no more instructions are available. This would be obvious for reasons for Claim 14.

28. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Joffe (US006330584B1) and Krishna (US006161173A) in view of Nguyen (US005961628A).

Claim 18 is similar in scope to Claim 8, and so is rejected under the same rationale.

Allowable Subject Matter

29. Claims 25-31 are allowed, for reasons given in the Office Action dated March 28, 2007.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of advisory action. In no event, however, will statutory period for reply expire later than SIX MONTHS from date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JONI HSU whose telephone number is (571)272-7785. The examiner can normally be reached on M-F 8am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kee Tung can be reached on 571-272-7794. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JH

/Kee M Tung/
Supervisory Patent Examiner, Art Unit 2628